

# Creating Robot Models for MPK - A Rough Guide

Sagar Behere  
behere@kth.se

August 2, 2009

## Abstract

This document summarizes the steps I took to create a robot model for the 6 DOF KUKA KR5 Sixx R850 robotic arm.

## 1 Introduction

Robot models for use with the MPK library need to have a “collision model” in Open Inventor’s .iv file format. These collision models are basically .iv models of individual robot links. The robot definition ( .rob ) file contains parameters needed to “assemble” the link models into the entire robot arm. The process of generating a suitable link model in .iv format and finding the “assembly” parameters is not immediately obvious. This document describes the steps I had to take to build a .rob file for my robotic arm.

## 2 Quick start

1. Generate CAD model of individual links
2. Open each model in blender 3d. Save it in blender format.
3. Shift origin of global coordinate system of each link to the point about which the link rotates
4. Export model obtained in step 3 to .iv format
5. Use blender to calculate offsets of coordinate system origin of each link from coordinate system origin of previous link.

## 3 Detailed description

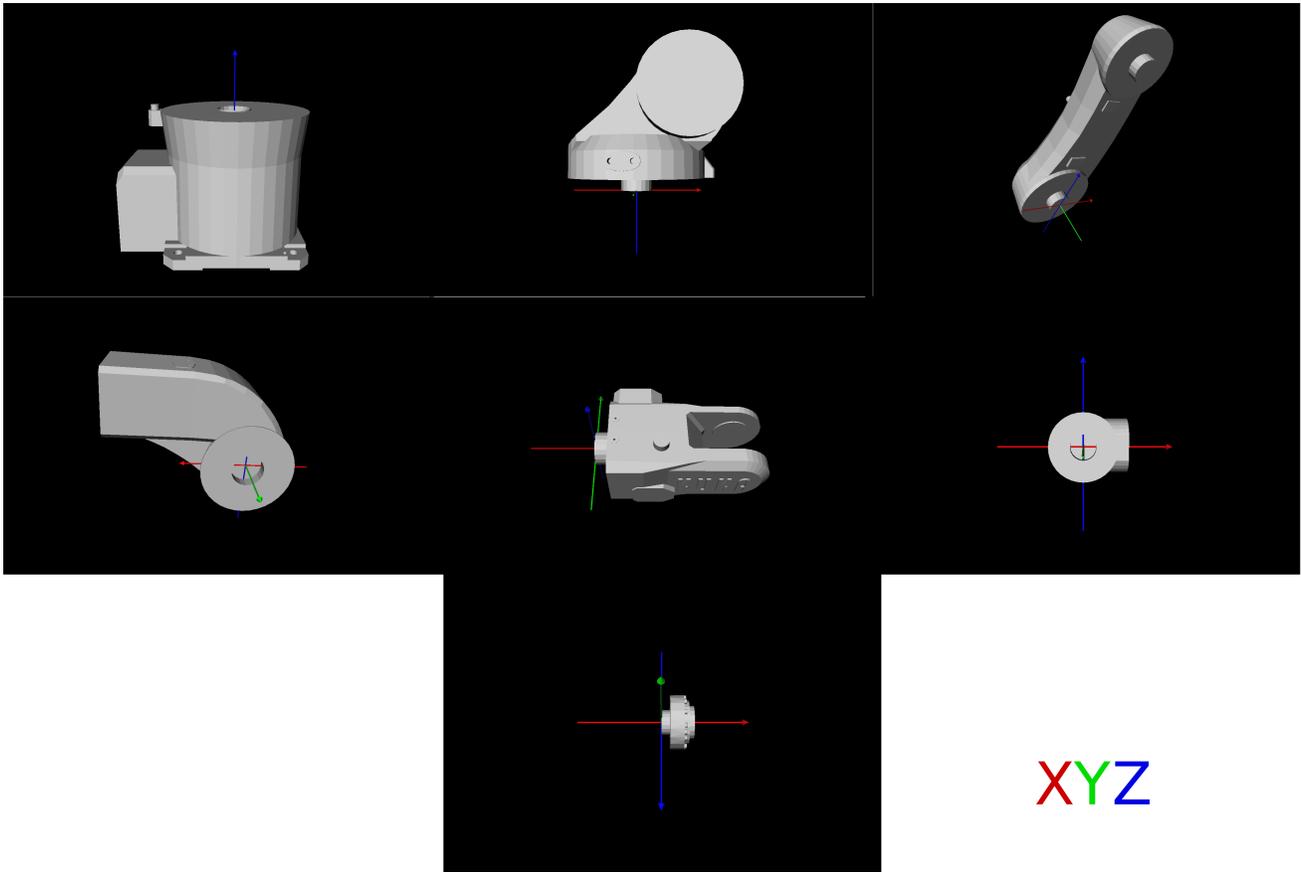
1. Download a CAD model of the robot from KUKA website in .STL format
2. Import the model into blender (<http://www.blender.org>). I used version 2.48a
3. Go to edit mode, select all vertices ('A' key), set the limit to 0 and remove doubles. This step makes it easier to select parts of the model which need to be deleted later. Also, it could lead to a smaller .iv format representation (less vertices)
4. Now, we need models for the individual links. Decide which link is needed and delete all other parts of the model. Save the file as a .blend file. Next, export it to the .iv format. View the resulting .iv file with the ivview program.
5. Repeat step 4 until you have .blend and .iv files for all links.
6. If you followed *exactly the same* steps while generating the link models for all links, the resulting .iv file models will be to scale and the individual link models will be positioned at the right places. This means that if you collect together all the .iv link models in a single .iv file (see box to see how), you’ll see the entire robot arm with all the links in their right places.

```
#Inventor V2.1 ascii
#robot model
Separator
{
File {name "kuka_0.iv"}
File {name "kuka_1.iv"}
File {name "kuka_2.iv"}
File {name "kuka_3.iv"}
File {name "kuka_4.iv"}
File {name "kuka_5.iv"}
File {name "kuka_6.iv"}
}
```

7. Next, we need to mess with the link models in order to position the local coordinate system (in the exported .iv files) at the right point and orientation. This is important because in the robot definition files, the joint rotations are specified around the axes of the local coordinate system. Unless the local coordinate system is in the right place in each link model, your robot links will go all over the place when you use the .rob file in MPK
8. The local coordinate system in the exported .iv file matches the global coordinate system in blender. Thus, you need to move the global coordinate system to that point on a link, about which the link rotates, when connected to the previous link. See figure 1 to see where I had to position the global coordinate system for each link of my robot. Once you have shifted the global coordinate system to the right place, save the .blend model and export it to .iv format again.
9. Now, if you use the code given in the box above to collect the links together into a robot model, its not going to work. This is because inventor will place all the links such that the origins of their local coordinate system coincide. You need to specify the offset of the origin of the coordinate system of each link, from the origin of the coordinate system of the previous link. For example, lets say we need to find the offset of link 3 w.r.t link 2. To do this, open the .blend model of link 2. Zoom in to the point on the link, where link 3's coordinate system origin will touch, when it is properly assembled. Get the coordinates of this point. These will be the offset of link 3 w.r.t link 2. Note this data carefully. You'll need it to build the .rob file.
10. Repeat the steps of point 9 till you know the offset of each link w.r.t its previous link.
11. With the data from point 10 and the .iv models from point 8 you can now assemble the coordinate system shifted .iv models into a complete robot model. The same data is used to create a .rob file. See files robot.iv and kuka.rob for the robot model and .rob file for my robot.

## 4 Appendix: Coordinate system visualization

It is often useful to insert 3 arrows (the X-, Y- and Z- axes) to depict the coordinate system in your .iv model. Use the code below in your .iv file to get an idea of where the origin lies and how the coordinate system is oriented.



XYZ

Figure 1: Link models

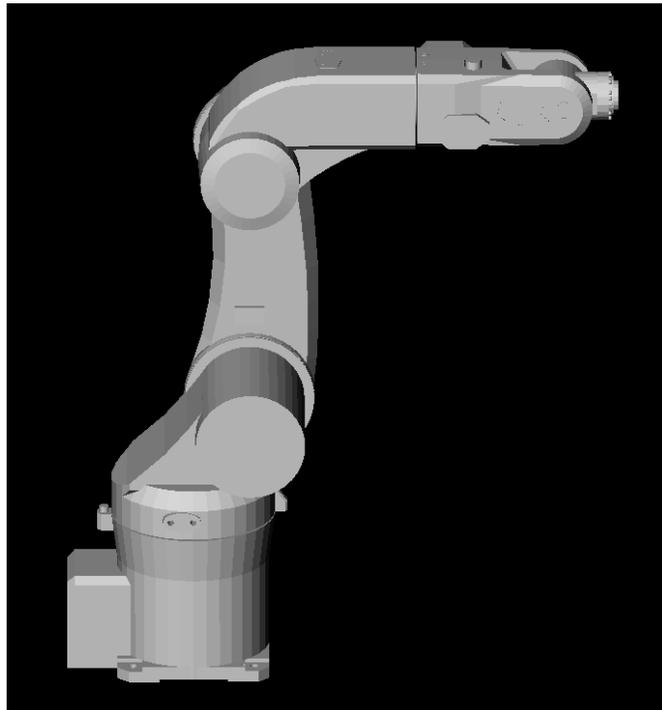


Figure 2: The assembled robot

```

#Coordinate system visualization
Separator
{
#Y-axis
Separator
{
    BaseColor {rgb 0 1 0}
    Rotation {rotation 0 0 0 -1.5707}
    Cylinder {
        parts    ALL    # SFBitMask
        radius  1      # SFFloat
        height  200    # SFFloat
    }
    Translation {translation 0 100 0}
    Cone {
        bottomRadius 4
        height 8
    }
}
#X-axis
Separator
{
    BaseColor {rgb 1 0 0}
    Rotation {rotation 0 0 -1 1.5707}
    Cylinder {
        parts    ALL    # SFBitMask
        radius  1      # SFFloat
        height  200    # SFFloat
    }
    Translation {translation 0 100 0}
    Cone {
        bottomRadius 4
        height 8
    }
}
#Z-axis
Separator
{
    BaseColor {rgb 0 0 1}
    Rotation {rotation 1 0 0 1.5707}
    Cylinder {
        parts    ALL    # SFBitMask
        radius  1      # SFFloat
        height  200    # SFFloat
    }
    Translation {translation 0 100 0}
    Cone {
        bottomRadius 4
        height 8
    }
}
}
}

```